

Prototype to obtain <keyword, answer> pairs on websites

Wang, Yun; RUID: 154000804

Hao Luo; RUID: 162006265

1 Objective of this project

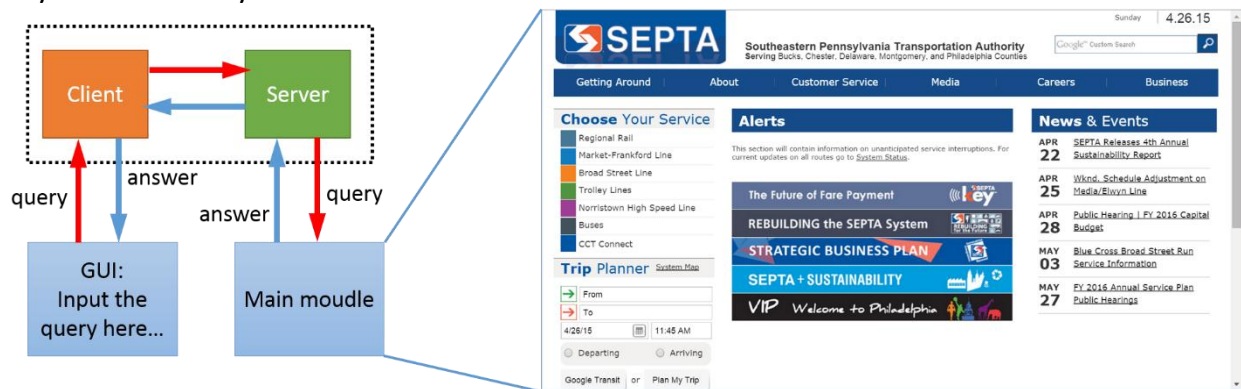
In this project we built a cloud-based and modular prototype to find (query kw, answer) pairs on a chosen website.

2 Description of our prototype

The prototype consists of three parts (scheme 1):

- 1) The front-end GUI used to input the query;
- 2) The communication module used to transmit query to server and fetch the answer from server;
- 3) Main module used to extract the target answer from webpage.

Almost all the works are done in module running in the server. The GUI and client modules are very small and easy to maintain.



Scheme 1. Structure of the prototype for obtaining <kw, answer> pairs on websites

3 Implement our prototype in septa (www.septa.org)

The website that we chose to implement our prototype is www.septa.org, which is a popular public transport website among the citizens of Philadelphia. The trip planner tool in this website could provide trip plan information to the customers. In order to obtain the travel plan information, customers should input the start station, the destination station, the date and time they want to travel, then the trip planner tool will search for their expected itineraries and show them to customers.

All of our programs were written with Python. Simulation process of our programs is shown as follows:

The server.py should be run first, then input query in GUI and click “Run” button, the programs will show target information automatically.

GUI

GUI used to input the query and convert the query to a string. It calls client module to transfer the string to server and present the target answer that client received.

The screenshot shows a window titled "SEPTA" with a red overlay on the left side. The overlay contains the following text:

- Starting Station
- Destination Station
- Schedule Time
- Schedule Data

The input fields in the window are:

- Starting Station: Claymont
- Destination Station: Trenton
- Schedule Time: 11:59PM
- Schedule Data: 04/30/2015

There is a "Run" button at the bottom left of the input area.

As shown in the figure above, a trip from Claymont to Trenton at 11:59PM on 4/30/2015 was input in GUI.

Client and Server

Client and server module were programmed based on socket. They established a TCP connection between client and server and transmitted information to each other.

```
*Python 2.7.8 Shell*
File Edit Shell Debug Options Windows Help
Python 2.7.8 (default, Jun 30 2014, 16:03:49) [MSC v.1500 32 bit (Intel)] on win
32
Type "copyright", "credits" or "license()" for more information.
>>> ===== RESTART =====
>>>
Claymont,Trenton,11:59PM,D,04/30/2015
ready for inputs

Format of input: From address,To address,Time,Depart by or Arrive by,Date
Example: Claymont,Trenton,05:00PM,D,04/30/2015
Claymont,Trenton,11:59PM,D,04/30/2015
Claymont,Trenton,11:59PM,D,04/30/2015
The best itinerary is:
  From Claymont to 30th Street Station
  From 30th Street Station to Trenton
Trip takes 1:49 And the fare is $8.50
The best itinerary is:
  From Claymont to 30th Street Station
  From 30th Street Station to Trenton
Trip takes 1:49 And the fare is $8.50
|
```

Server

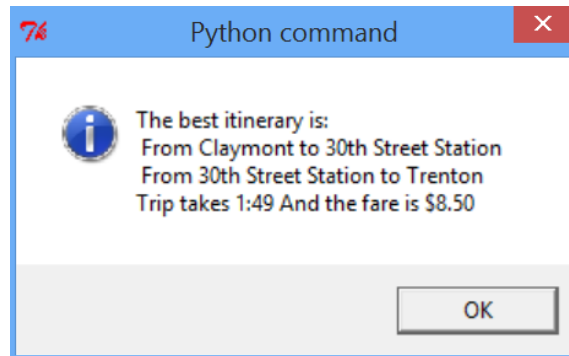
Server called the septa module and transmitted the resulting answer to client.

```
*Python 2.7.9 Shell*
File Edit Shell Debug Options Windows Help
Python 2.7.9 (default, Dec 10 2014, 12:24:55) [MSC v.1500 32 bit (Intel)] on win
32
Type "copyright", "credits" or "license()" for more information.
>>> ===== RESTART =====
>>>
Ready for connection
...connected from ('172.31.83.155', 58494)
Ready for connection
|
```

Main module: Septa

Main module septa used a `selenium.webdriver()` to transfer the query to webdriver and extracted the target answer from the html of the website.

4 Simulation Results



This result tells that one could take train from Claymont station to the 30th Street Station and transfer to another train from 30th Street Station to Trenton station. This trip will take at least 1 hour and 49 minutes and cost is \$8.5.

5 Conclusion

We built a cloud-based and modular prototype to find the <kw query, answer> pairs on website. Then we implemented our prototype on www.septa.org and successfully got the target answers of queries. In our prototype, almost works are done on the remote server, so the GUI and client running locally are small and easy to maintain.